MSc Project Report

Simplified AI Assistant Robot Using Arduino and Python

Report by
Tayyab Rehman

Supervisor
Kaung Oo Htet

Date
28/04/2025

SCHOOL OF ENGINEERING AND COMPUTER SCIENCE
UNIVERSITY OF HERTFORDSHIRE
www.herts.ac.uk

PLAGIARISM STATEMENT

I have read the detailed guidelines to Students on academic offenses and misconduct information (Refer to the Ask Herts web page for the Assessment Offences and Academic Misconduct).

I have read and understood the information posted on Canvas Project module page about "Academic integrity, misconduct, plagiarism and unauthorised use of Artificial Intelligence (AI) in researching or presenting material for an assessment". (Click this Ask Herts link for more information)

I have attended the mandatory seminar delivered by SASH and/or the School Academic Integrity Officer (SAIO) about Assessment Offences and Academic Misconduct.

I understand the University has zero tolerance in academic offenses and misconduct, any suspected case may be subject to an investigation by the SAIO and disciplinary penalties may apply if the misconduct case is confirmed. (Reference: University Policy Regulation AS14, Appendix III – version 17.0)

Hereby, I certify that the work submitted is my own and that any material derived from AI or quoted from any the published or unpublished work of other persons has been duly acknowledged.**)**

Signature:

Student name: Tayyab Rehman
SRN: 23013956

UNIVERSITY OF HERTFORDSHIRE
SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

In this project, a wheelchair robot that can be operated with voice commands and physical controls is developed and controlled through an Android application. An Arduino Uno is used in the system's construction for motor control, and Bluetooth is used to connect to the mobile application. By using the Google API to implement voice recognition, users can control the wheelchair hands-free by speaking commands. An additional control technique for more versatility is manual directional control using on-screen buttons. The report addresses important technical issues and assesses system performance while going into detail about the system's design, integration, and testing. The results demonstrate how well voice technology and mobile applications work together to improve assisted mobility solutions..

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Kaung Oo Htet, for their invaluable guidance, encouragement, and constructive feedback throughout this project. Their insights and expertise have been instrumental in shaping the direction of my work.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AI	Artificial Intelligence
Lidar	Light Detection and Ranging
PWM	Pulse Width Modulation
SLAM	Simultaneous Localization and Mapping
IR Sensors	Infrared Sensors
PID Controller	Proportional-Integral-Derivative Controller
RTOS	Real-Time Operating System
NLP	Natural Language Processing
IoT	Internet of Things
PWC	Power Wheelchair
PWM Signal	Pulse Width Modulation
ToF	Time-of-Flight

Chapter 1

INTRODUCTION

The need for sophisticated healthcare and assistive technologies is rising as a result of the world's population growth. People with mobility disabilities frequently have trouble navigating and being independent, especially in crowded or difficult surroundings. People with severe disabilities may find it challenging to operate traditional wheelchair systems manually. It is now easier to include AI-powered automation into mobility solutions because to continuous developments in embedded systems and artificial intelligence. A hands-free option is provided by voice-controlled wheelchairs, which enable users to manoeuvre with ease using straightforward speech commands. However, precision, avoiding obstacles, and price are frequently issues with current options.

The goal of this project is to create a voice-activated, AI-powered wheelchair system that will increase the mobility of people with disabilities. The Raspberry Pi is used for obstacle avoidance and navigation, while DFRobot Voice Recognition is used for command input. This project aims to develop an effective, dependable, and reasonably priced assistive solution for those with restricted mobility by integrating real-time voice recognition and autonomous navigation.

Figure 1.1 AI-driven, voice-controlled wheelchair [1]

1.1 Aims and Objectives

1.1.1 Aim:

To design and prototype a voice-controlled robotic assistant that uses Arduino and Python to perform simple tasks, demonstrating the potential of integrating AI into robotic systems..

1.1.2 Objectives:

1. Implement a Voice Recognition System: Use Python to incorporate speech-to-text functionality that translates voice commands into actionable tasks.

2. Develop Robotic Control- Program an Arduino to respond to the processed commands by performing predefined actions.
3. System Integration- Merge the voice recognition software with the hardware controls to create a unified robotic assistant.
4. Testing and Refinement- Evaluate the system's responsiveness and accuracy in real-world scenarios and refine based on feedback

1.2 Project Specification

Based on the above discussion, it is necessary to discuss the key components of the proposed voice-controlled AI assistant wheelchair. The project consists of several integral parts, which are described below:

Table 1.1 Project Specification

Sr No.	Components	Specification
1	Wheelchair Type	Motorized, Voice-Controlled
2	Embedded Controller	Arduino, Raspberry Pi
3	Voice Recognition Module	DFRobot Voice Recognition V3
4	Motor Speed Controller	H-Bridge Motor Driver
5	Motor Assembly	DC Motors with PWM Speed Control
6	Navigation System	Ultrasonic Sensors, IR Sensors
7	Communication Channel	Wi-Fi, Bluetooth
8	AI Processing	Python-based Speech-to-Text

1.3 Project Plan

This project is an individual MSc research project focused on developing a Simplified AI Assistant Robot using Arduino and Python, integrating a voice-controlled wheelchair with DFRobot Voice Recognition and LiDAR-based navigation using Raspberry Pi. The project is scheduled to be completed by May 5, 2025, with systematic progress over the weeks.

The project involves research, hardware assembly, software integration, testing, and final prototype validation.

1.3.1 Project Milestones

The table below presents the distribution of tasks, duration, and responsible individual (**Tayyab Rehman**).

Table 1.2 Project Milestones

Sr No.	**Task**	**Duration**
1	Literature Review	1 Week
2	Selection of Components (Microcontroller, Motors, Sensors, Power Supply)	1 Week
3	Assembling the Wheelchair Frame and Actuators	1 Week
4	Interfacing motor driver with arduino	1 week
5	Interfacing DFRobot Voice Recognition Module With Ardiuno	1 Weeks
6	Testing Voice Commands and Motion Control	2 Week
7	Developing Navigation System for the Voice-Controlled Wheelchair using Lidar	3 Weeks
8	Integrating Arduino with Raspberry Pi for Control System	2 Weeks
9	Interim Report	5 Week
10	Dissertation	7 weeks

1.3.2 Project Timeline

The project timeline has been provided in the form of a Gantt chart here each activity is provided along with the respective time in which it is completed.

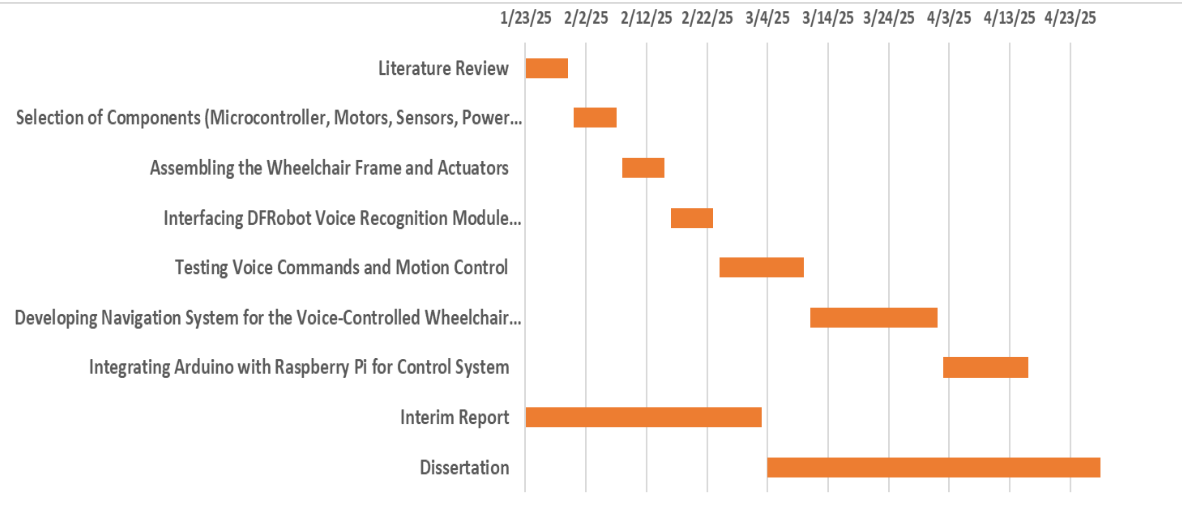

Figure 1.2 Project Timeline

1.4 Ethical Implications

Ethical considerations in this project are critical, as it involves user interaction and data processing. Key ethical aspects include:

- **User Privacy:** Protecting user privacy by making sure voice data is processed locally without storing sensitive information..
- **Bias and Inclusivity:** The voice recognition system needs to be inclusive in order to correctly identify commands from users with various speech patterns or accents..
- **Accessibility and Usability:** Creating a user-friendly interface that can be used by people with different levels of cognitive and mobility abilities.
- **Transparency and Accountability:** Making sure users are aware of the system's risks and capabilities while also clearly communicating its limits.

1.5 Project Risk

Potential risks associated with the project include:

- **Hardware Failures:** Operational problems caused by motor, sensor, or voice recognition module malfunctions.
  - *Mitigation:* Keep backup parts on hand and carry out complete testing..
- **Software Bugs and Integration Challenges:** Problems with Arduino, Raspberry Pi, and Python module compatibility.
  - *Mitigation:* Use strict debugging procedures and modular code techniques..
- **Voice Recognition Accuracy:** speech commands being misunderstood, particularly in noisy settings..
  - *Mitigation:* Optimize speech-to-text algorithms and make use of noise-cancelling microphones.
- **Delays and Resource Constraints:** Possible delays brought either by technical difficulties or component unavailability.
  - *Mitigation:* Assign a buffer period in the timeline and make advance plans with alternate providers.

1.6 Capital Resources

A detailed budget is essential for efficient resource allocation. The estimated costs for each component are as follows:

Table 1.3 Project Budget

Sr No.	Component/Resource	Estimated Cost (£)
1	Wheelchair Frame and Motors	14
2	DFRobot Voice Recognition Module	18
3	Arduino	25
4	Raspberry Pi	34
5	H-Bridge Motor Driver and Power Supply	20
6	Miscellaneous Components (Wiring, Connectors, Mounts)	20
Total Estimated Cost		£131

Chapter 2

LITERATURE REVIEW

The research on voice-controlled wheelchairs and their components—such as the motor assembly, embedded controllers, voice recognition modules, structural design, and navigation systems—is examined in this chapter. In addition, it looks at comparable projects and technologies, pointing out the shortcomings of current solutions and outlining the study's problem statement.

2.1 Overview of Voice-Controlled Wheelchairs

A wheelchair is a type of mobility aid intended to assist those who are unable to walk or who struggle to walk because of a disability, disease, or accident. With two bigger rear wheels and two smaller front wheels for stability and manoeuvrability, it usually has four wheels, a seat, and footrests. For increased mobility and independence, wheelchairs can be driven by electric motors or manually operated by the user or an aid. Manual wheelchairs can be pushed by an attendant pushing from behind or by the user using hand rims attached to the back wheels. On the other hand, rechargeable batteries power electric wheelchairs, which are operated by a joystick or other input devices. This makes it easier to navigate on different terrains and enables precision movement. In everyday life, sports, leisure activities, and healthcare settings, wheelchairs are frequently utilised. Because of the intricate dynamics involved in ensuring safe and comfortable motion, they have drawn a lot of interest from researchers and engineers. To accommodate users with varying physical abilities, modern wheelchair designs integrate a variety of control technologies, such as voice control, joystick controllers, and brain-computer interfaces. For autonomous mobility and obstacle avoidance, sophisticated navigation systems utilising sensors like gyroscopes, accelerometers, GPS, sonar, and laser sensors are also being incorporated. To improve stability, safety, and user experience, a variety of control strategies have been investigated, such as PID controllers, fuzzy logic, and machine learning algorithms. While taking social and environmental factors into account, the ongoing development of intelligent wheelchairs seeks to increase users' independence and accessibility. [2]

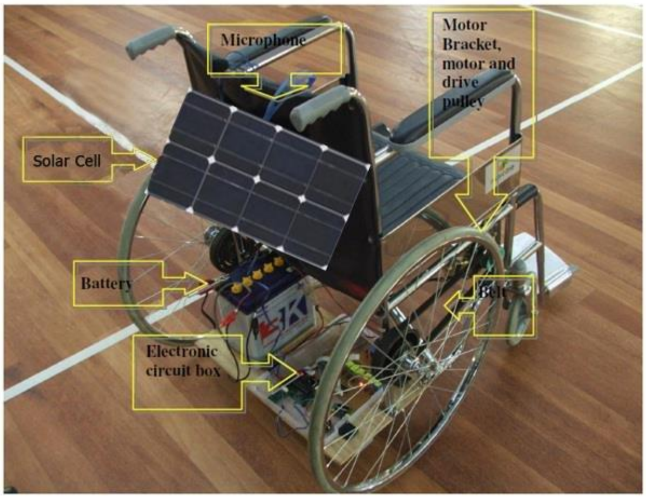

Figure 2.1 Voice Controlled Wheelchair [3]

2.1.1 Types of Wheelchair

Wheelchairs come in various types, each designed to meet specific user needs, preferences, and environments. They can be broadly categorized into the following types: [4]

2.1.1.1 Manual Wheelchairs

The most basic and conventional kind of wheelchairs are manual wheelchairs. Wheels must be propelled forward or backward by the user's own upper body strength and movement; they are not electrically or battery-powered. If the user is unable to roll themselves, a carer can push and manoeuvre this kind of wheelchair. [5]

2.1.1.2 Power Wheelchairs

Power wheelchairs are completely automated, typically battery-operated, and ideal for people who lack the muscle or dexterity to operate a manual wheelchair. They are frequently far more robust and supportive than manual wheelchairs and come with joysticks and other basic controls for movement and manoeuvring. On the down side, though, this does make them considerably bigger and heavier. [5]

2.1.1.3 Sport Wheelchairs

Sport wheelchairs are extremely easy to manoeuvre and nearly impossible to topple because to their cutting-edge design. Professional athletes and athletes of all skill levels use them because they offer speed, fluidity, and movement in a stable and robust framework. [5]

Figure 2.2 Types of Wheelchair [5]

2.2 *Building blocks of Modern wheelchair*

Modern wheelchairs have evolved significantly from their traditional manual counterparts, incorporating advanced technologies to enhance mobility, comfort, and independence for users. This section delves into the key components that constitute the building blocks of modern wheelchairs, including structural design, motor assembly, drive systems, voice recognition, embedded controllers, and navigation systems.

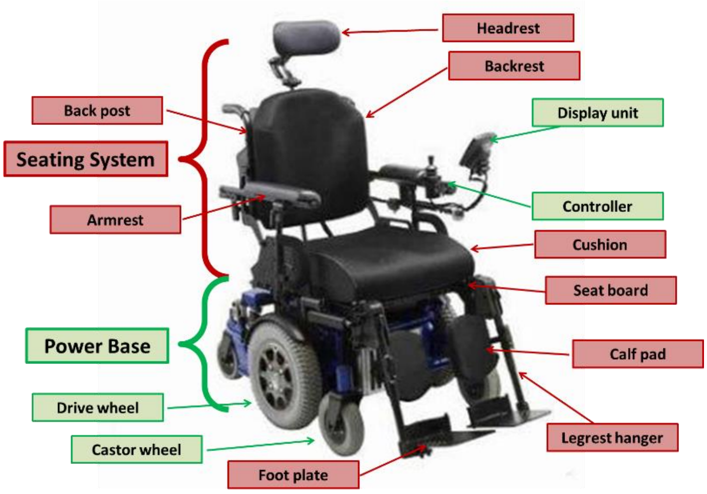

Figure 2.3 Power Base Wheelchair [6]

2.2.1 Structural Design and Materials

Modern wheelchairs are built with longevity, weight reduction, and ergonomics in mind. In order to provide strength without adding undue weight, lightweight materials like carbon fibre, titanium, and aluminium alloys are frequently utilised to make the frame. Additionally, these materials resist wear and corrosion, increasing the wheelchair's lifespan. [7]

2.2.2 Drive System

A power wheelchair's (PWC) drive system consists of the controller, motors, electrical modules, and auxiliary hardware and software. Together, these elements give the user movement and usefulness.. [7]

2.2.3 Controller

The controller allows the user or carer to steer and operate power seating functions. It comes in two types: [7]

- **Proportional Controllers:** These consist of attendant controls and joystick-style controllers that offer smoothly adjustable speed and direction.
- **Non-Proportional (Digital) Controllers:** These include directional control systems with little speed change, such as "sip and puff" or head arrays.

2.2.4 Motors

The chair is driven and steered by motors that provide sufficient torque to negotiate difficult terrain while adhering to speed limits. Users with restricted hand function benefit from advanced steering systems that assist them maintain a straight track over cambered ground. For added safety, motors include automated braking systems. [7]

2.2.5 Electronic Modules

PWCs use modules to manage drive response, speed, acceleration, and additional functionalities like Bluetooth or infrared connectivity: [7]

- **Control Module:** Enables performance changes and converts user inputs into motor actions.
- **Power Module:** Provides power to the motor for steering and drive. .
- **Multi-Actuator Module:** Necessary for a variety of power sitting tasks..
- **Environmental Control Module:** Allows for appliance communication

2.2.6 Seating System

The user is supported by the sitting system's cushions, leg rests, armrests, backrests, and postural supports. It may be adjusted to control pressure and posture. Power seating features improve usability and comfort, including [7]

- **Tilt-in-Space:** Helps with stability and pressure management by adjusting the seat angle without altering the seat-to-back angle.
- **Seat Elevation:** Enhances functional reach and social interaction while regulating pace for safety.
- **Power Backrest Recline:** Alters the angle of support from the seat to the back, however anti-shear devices might be necessary to stop sliding..
- **Largest Elevation:** Supports comfort and pressure control by adjusting the angle of the seat to the leg rest..

2.3 Navigation Mechanism

The approach known as Simultaneous Localisation and Mapping (SLAM) allows autonomous robots to map an unfamiliar area while also figuring out where they are on that map. Because LiDAR (Light Detection and Ranging) is so accurate at detecting distances and creating intricate point clouds of the surroundings, it is frequently utilised for SLAM. A spinning LiDAR scanner, which shoots laser rays at various angles to detect objects in the surroundings, is used to acquire point cloud data at the start of the mapping process. Each point in the cloud is represented as a three-dimensional tensor [8]

$$\Lambda_{n.a} = \Lambda(r_{n.a}, \psi_n, \alpha_a) \text{ [8]}$$

$r_{n.a}$ is the distance computed using the Time-of-Flight (ToF) method

ψ_n is the elevation angle

α_a is the azimuth angle

The transformation from spherical coordinates to Cartesian coordinates is given by:

$$\begin{bmatrix} \cos\alpha_a & \cos\psi_n \\ \sin\alpha_a & \cos\psi_n \\ \multicolumn{2}{c}{\sin\psi_n} \end{bmatrix}$$

This equation converts LiDAR points into a rectangular (Cartesian) coordinate system, which forms the basis for mapping. Histograms of the point cloud data are used to identify landmarks, like the corners of walls. The following formula is used to find the Most Significant Corner (MSC): [9]

$$C = min(H_x(y_c), H_y(X_c))$$

2.4 Related Projects

There are some related projects which are discussed below:

2.4.1 Voice Controlled Intelligent Wheelchair

In this work, a voice-controlled intelligent wheelchair that uses the grammar-based recognition parser "Julian" to enable control using Japanese speech commands is developed to help people with physical disabilities. To guarantee precise and secure wheelchair operation, the system makes use of three different kinds of commands: verification, short movement, and fundamental reaction commands. High speech recognition rates of 98.3% and 97.0% for movement and verification orders, respectively, were shown in experiments involving 15 individuals. Tested in a variety of settings, such as university hallways and cramped interior areas, the device demonstrated its efficacy and usefulness while demonstrating its potential to improve mobility for individuals with physical constraints. [10]

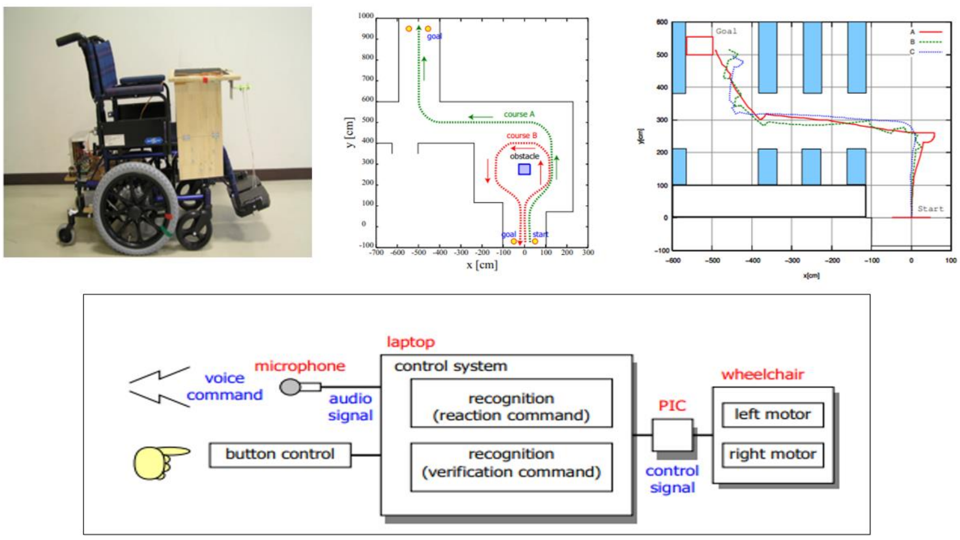

Figure 2.4 Voice Controlled Intelligent Wheelchair [10]

2.4.2 Arduino based voice controlled wheelchair

This study describes the creation of a voice-activated wheelchair that uses Arduino technology to help people with limitations of the upper and lower limbs. To control the wheelchair's movement through motor controllers, the prototype incorporates a speech recognition module that translates vocal commands into signals that are subsequently processed by an Arduino microcontroller. Messy wiring is not necessary for wireless connection thanks to a Bluetooth module. The system has a good recognition accuracy and is reasonably priced. It can recognise voice commands in English, Chinese, and Malay. It also has infrared sensors for obstacle detection and an optional joystick for manual control. With a voice recognition success rate of 96–100% in all three languages, the prototype underwent successful testing. [11]

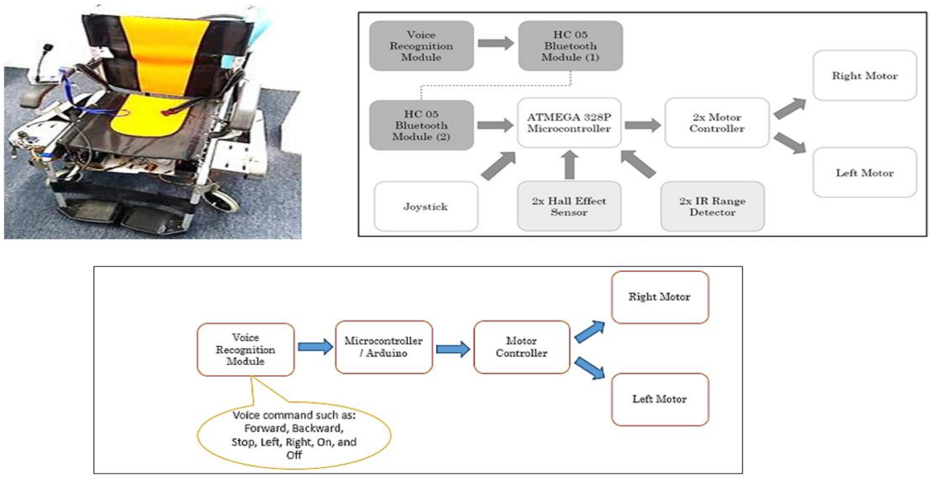

Figure 2.5 Arduino Based Voice Controlled Wheelchair [11]

2.5 Limitations and Bottlenecks of the Existing Work

While significant progress has been made in developing voice-controlled intelligent wheelchairs, several limitations and challenges remain in the existing systems:

- The Arduino-based system detects obstacles using infrared sensors, which are less accurate and have a shorter range than more sophisticated sensors like LIDAR or ultrasonic sensors. In dynamic areas with moving impediments, this could jeopardise safety..
- Both systems were limited to user-directed movement because they lacked autonomous path planning and sophisticated navigation algorithms. The user experience could be greatly improved by including AI-based navigation, such as machine learning-based obstacle avoidance or SLAM (Simultaneous Localisation and Mapping).

2.6 Analysis Procedure

Before starting the project Analysis of different strategies and approaches that can be applied to make this project were analysed and based totally on this analysis most appropriate approach and technique is selected for the project implementation.

2.6.1 Microcontroller Comparison

To select the most suitable microcontroller for this project, a detailed comparison of different options was conducted, focusing on features such as processing power, memory, communication interfaces, and power consumption. The following table summarizes the key features of the most commonly used microcontrollers:

Table 2.1 Microcontroller Comparison

Feature	Arduino Uno	ESP32	Raspberry Pi 4	STM32
Architecture	8-bit AVR	32-bit Xtensa LX6	64-bit ARM Cortex-A72	32-bit ARM Cortex-M3
Clock Speed	16 MHz	240 MHz (dual-core)	1.5 GHz (quad-core)	72 MHz
Flash Memory	32 KB	4 MB (varies by model)	MicroSD storage	64 KB
RAM	2 KB	520 KB	2 GB / 4 GB / 8 GB LPDDR4	20 KB
Processing Power	~20 MIPS	~600 DMIPS	~13,000 DMIPS	~90 MIPS
Power Consumption	~50 mA @ 5V	5-240 mA (depends on mode)	~600-800 mA @ 5V	~30-40 mA @ 3.3V
GPIO Pins	14 digital, 6 analog	34 (varies by model)	40 (multipurpose)	37
PWM Outputs	Yes	Yes	Software-based	Yes
Analog Inputs	6	18	No (uses external ADC)	10
Communication	UART, I2C, SPI	UART, I2C, SPI, CAN, Ethernet	UART, I2C, SPI, Ethernet, HDMI, USB	UART, I2C, SPI
WiFi / Bluetooth	No	Yes (WiFi + BLE)	Yes	No

Operating System	None (Bare Metal)	None (Bare Metal, FreeRTOS)	Linux-based (Raspberry Pi OS, Ubuntu)	None (Bare Metal, RTOS)
Ease of Use	Very beginner-friendly	Moderate (requires FreeRTOS, WiFi setup)	Complex (requires Linux setup)	Moderate (STM32CubeIDE required)
Best For	Simple projects, learning	IoT, AI on edge, wireless applications	AI, ML, robotics, full applications	Real-time applications, motor control
Cost	£10-15	£7-12	£40-60	£5-10

The **Arduino Uno** was chosen for motor control and command processing because of its dependability, simplicity, and ease of interaction with the DFRobot Voice Recognition Module and motor drivers.

Because of its strong processing capabilities—which are critical for processing Lidar data in real-time and advanced navigation algorithms—the **Raspberry Pi 3** was selected for navigation and obstacle avoidance.

2.6.2 Evaluation of Voice Recognition Methods

Before initiating the project, various strategies and approaches for implementing voice recognition were evaluated. The aim was to determine the most effective solution by considering factors such as cost, ease of integration, functionality, and specific project needs. Following this evaluation, the optimal method was chosen for the voice-controlled wheelchair. The table below provides a comparison of the different voice recognition options considered:

Table 2.2 Voice Recognition Methods [12]

Voice Recognition Method	Pros	Cons	Cost	Best Use Case
EasyVR Shield	Offline, easy to integrate with Arduino	Limited vocabulary, requires training	~£40	**Best for** basic offline control with Arduino
DFRobot Voice Recognition	Offline, supports multiple commands	Limited to pre-defined commands	~£25	**Good for** predefined commands,

				simple projects
Google Assistant API	Advanced NLP, supports natural language	Needs internet, privacy concerns	Free (cloud-based)	**Best for** AI-powered control if internet is available
Alexa Voice Service	Advanced NLP, supports natural language	Requires internet, complex integration	Free (cloud-based)	**Best for** smart home & cloud-based AI control
ESP32 + TinyML	Offline, customizable, lightweight	Requires training & ML expertise	~£8-10	**Best for** custom voice commands & ML-based solutions

After careful consideration, the DFRobot Voice Recognition Module was chosen for this project due to its offline capability, cost-effectiveness, and ease of integration with Arduino. It allows for multiple predefined commands, making it suitable for the wheelchair's basic movement controls. Additionally, its reliable performance in offline mode ensures continuous operation without the need for an internet connection, enhancing user safety and convenience.

2.6.3 Selection of Motor Driver

For this project, the L298N Dual H-Bridge motor driver was chosen to control the BO DC motors of the voice-controlled wheelchair. This motor driver is cost-effective and easily integrates with the Arduino Uno, making it suitable for basic motion control applications.

Table 2.3 Comparison of Motor Drivers

Motor Driver	Pros	Cons	Cost
L298N Dual H-Bridge	Cheap, easy to use with Arduino	Not efficient for high power motors	£3-8
BTS7960B (43A)	High power, suitable for wheelchairs	Slightly expensive	£10-20
Sabertooth 2x25A	High performance, robust	Expensive	£80-100

Chapter 3

PROJECT DESIGN AND IMPLEMENTATION

For the design and implementation of this voice-controlled wheelchair, multiple techniques were used to complete the project. All the methodologies, hardware components, and the programming flowchart involved in the development of this system are explained below:

3.1 *Proposed Design Methodology*

The main blocks of this project are given below in the block diagram. Each block is defined to clarify the proposed design methodology.

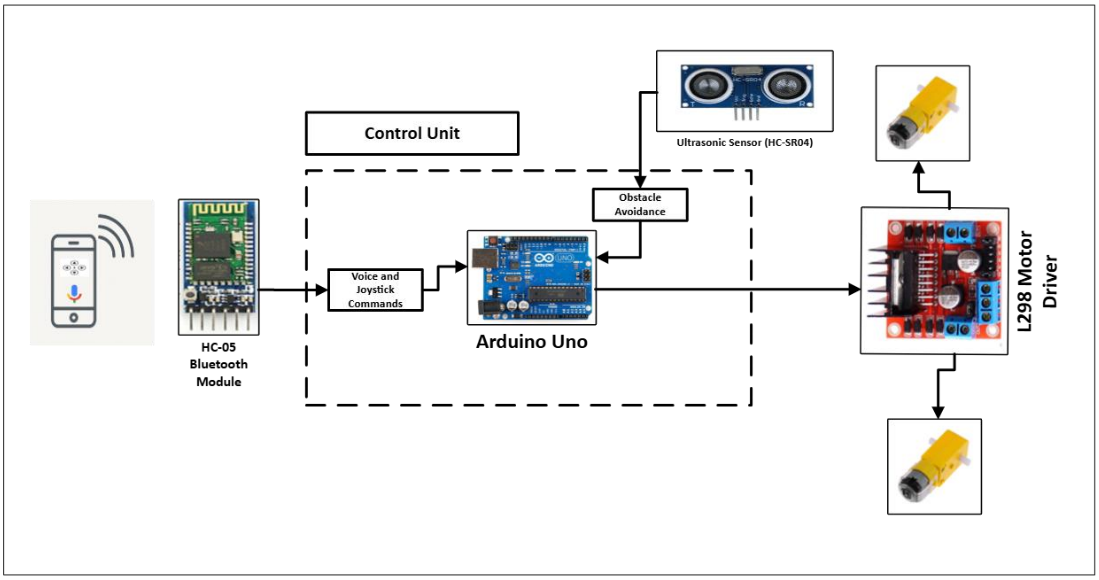

Figure 3.1 Block Diagram of Project

3.1.1 Arduino Uno

The Arduino Uno microcontroller acts as the central control unit. It receives control commands (both voice and joystick-based) from the Android app via Bluetooth, processes them, and sends corresponding signals to the motor driver for motion control. It also continuously monitors the sonar sensor to avoid collisions.

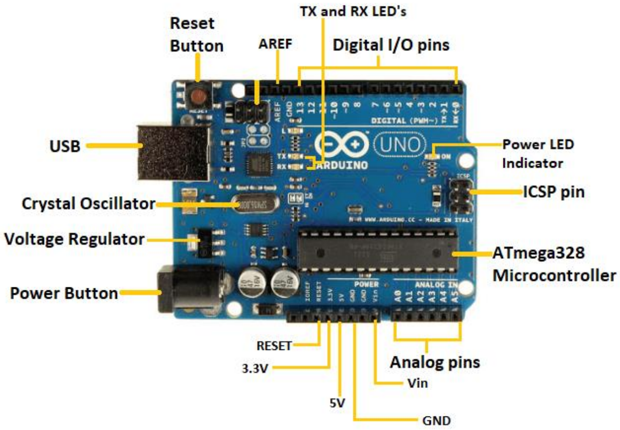

Figure 3.2 Arduino Uno [12]

3.1.2 Motor Driver

By connecting the DC motors to the Arduino microcontroller, the motor driver in this project is essential to managing the wheelchair's motion. The motor driver serves as a link between the high-power needs of the motors and the Arduino's low-power control outputs. By modifying the voltage and current given to the motors, it permits exact control over speed and direction. It also facilitates bidirectional mobility, which makes it possible for the wheelchair to turn, travel ahead, and move backward effectively. In order to ensure smooth and dependable motion control, the motor driver employed in this system was chosen based on its compatibility with the chosen DC motors and its capacity to manage the necessary power levels.

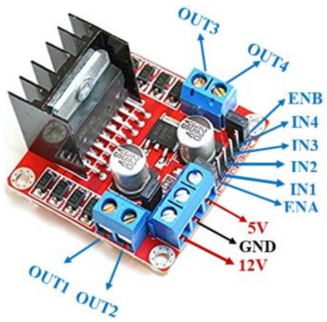

Figure 3.3 L298N Motor Driver [13]

Table 3.1 L298N Module Pinout Configuration [15]

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B

OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

3.1.3 Dc Motors

The voice-controlled wheelchair's wheels are driven by BO DC motors in this project, enabling dependable and effective mobility. Because of their small size, low weight, and affordability, BO (Battery Operated) DC motors are well-suited for robotics and mobility applications. The wheelchair accelerates and decelerates smoothly because to these motors' sufficient torque and speed. Additionally, they facilitate bidirectional rotation, which makes it easy for the wheelchair to move forward, backward, and turn. Furthermore, BO DC motors may be readily controlled with Arduino's Pulse Width Modulation (PWM) signals, which allow for accurate speed and direction control. They are also compatible with motor drivers. Longer battery life is ensured by their low power consumption, which also adds to the wheelchair's overall energy efficiency.

Figure 3.4 BO DC motor [14]

3.1.4 Ultrasonic Sensor (HC-SR04)

An HC-SR04 ultrasonic sensor is used for obstacle detection. It continuously scans the front environment and sends distance measurements to the Arduino. If an object is detected within a threshold (e.g., <30 cm), the system automatically stops or overrides unsafe commands.

Figure 3.5 Ultrasonic Sensor (HC-SR04) [15]

3.1.5 HC-05 Bluetooth Module

The HC-05 module enables wireless communication between the Android app and the Arduino. It receives data packets containing joystick directions or voice command keywords, which are then interpreted by the Arduino to perform corresponding movements.

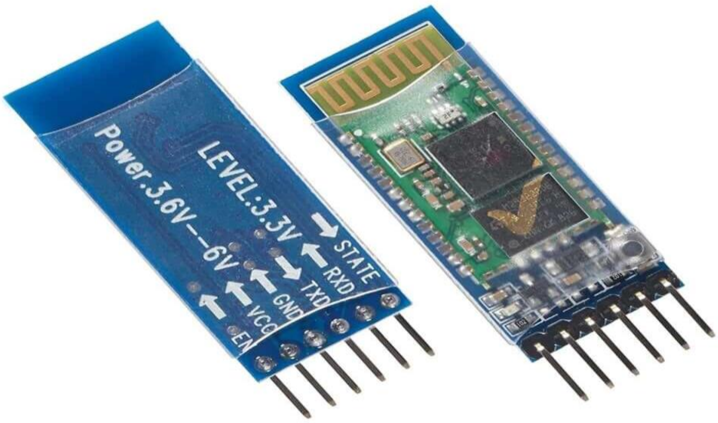

Figure 3.6 HC-05 Bluetooth Module [16]

3.1.6 Android Application

A user-friendly Android application was developed using Android Studio with Java to provide an intuitive control interface for the wheelchair. The app integrates two primary control modes: a voice command interface and a joystick interface. The voice command feature utilizes Android’s built-in speech recognition to convert spoken words into text and transmits the corresponding commands to the wheelchair via Bluetooth. Commands such as “forward,” “left,” or “stop” are interpreted and sent in a simplified string format like "F" for forward or "L" for left. In parallel, the joystick interface enables real-time manual control, allowing the user to adjust direction and movement dynamically with their thumb. The application also includes a connection status indicator, ensuring the user is informed when a successful Bluetooth pairing with the HC-05 module is established. These features collectively enhance accessibility and ease of use, offering both hands-free and manual control options for the wheelchair.

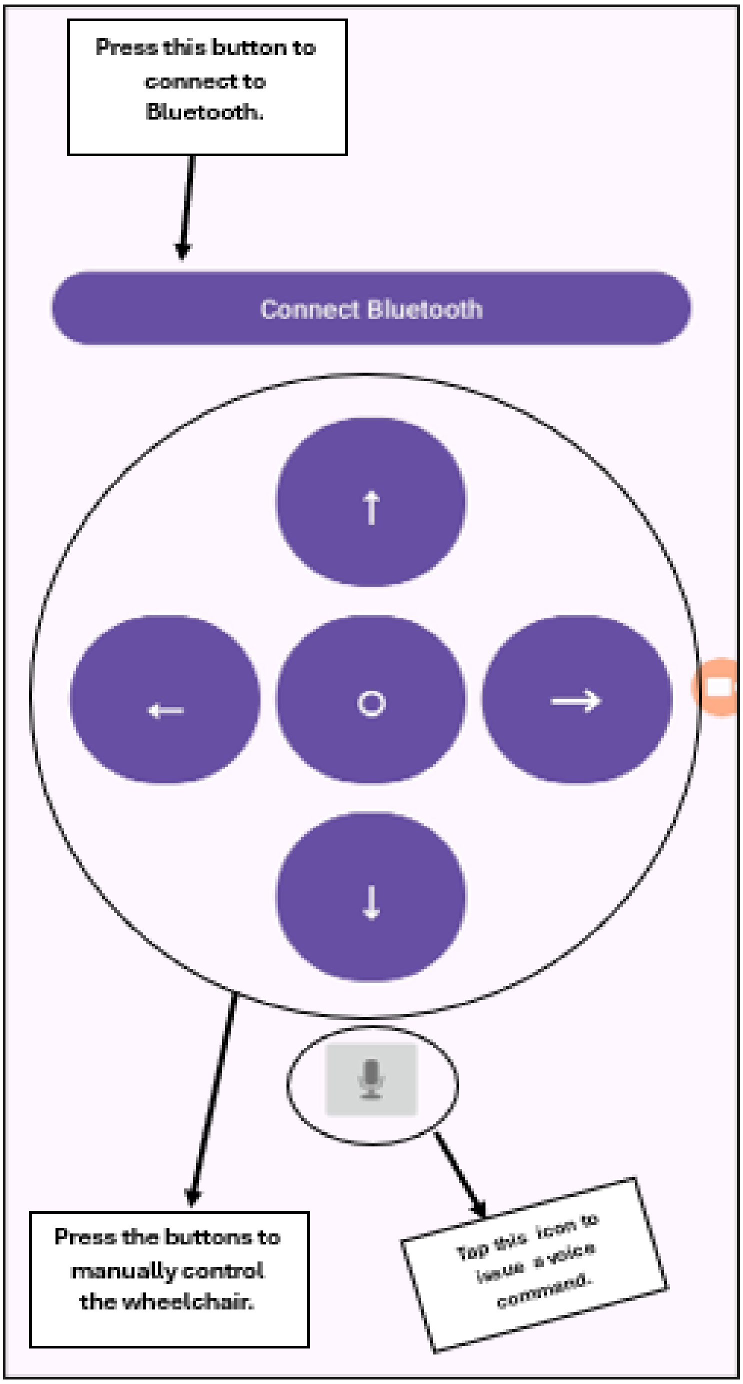

Figure 3.7 Android Application

3.1.7 Wheelchair base

The foundational structure of the voice-controlled wheelchair system is built on a two-wheel drive (2WD) robot chassis, which serves as the physical base of the project. This chassis was selected for its compact design, ease of assembly, and suitability for indoor and flat-surface navigation.

The chassis is made from durable and lightweight materials such as acrylic or aluminium, providing enough strength to support all mounted components, including motors, the control board, and power supply, while maintaining low weight to ensure efficient motor operation.

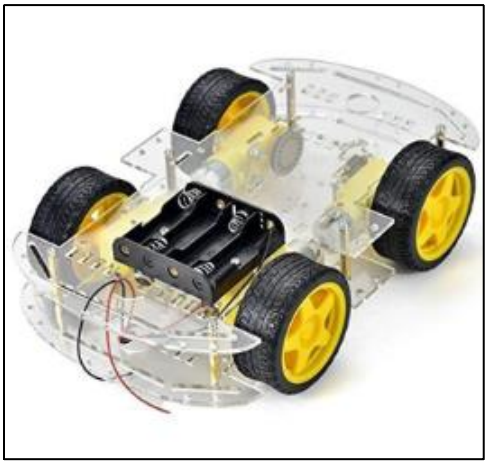

Figure 3.8 Robot Chassis Base [17]

3.2 *Implementations Strategy*

3.2.1 Arduino Programming And Logic Testing

Using the Arduino IDE, code is written and uploaded to the Arduino at this step. Managing input from the Bluetooth module and adjusting the motors appropriately are part of the logic. To guarantee precise replies to orders, extensive testing is carried out. The Serial Monitor is used for debugging in order to confirm that the Arduino understands and carries out commands as intended.

Figure 3.9 Arduino IDE

The wheelchair control system's general logic is depicted in the flowchart. It demonstrates how the system boots up, establishes a Bluetooth connection, interprets user input from voice commands or direction buttons, and transmits the proper control signals to drive the wheelchair. Real-time manual or voice-based control is possible without termination since the main program loop continuously checks for user actions. Standard flowchart symbols are used to appropriately depict input and output procedures and reflect the functionality of the system.

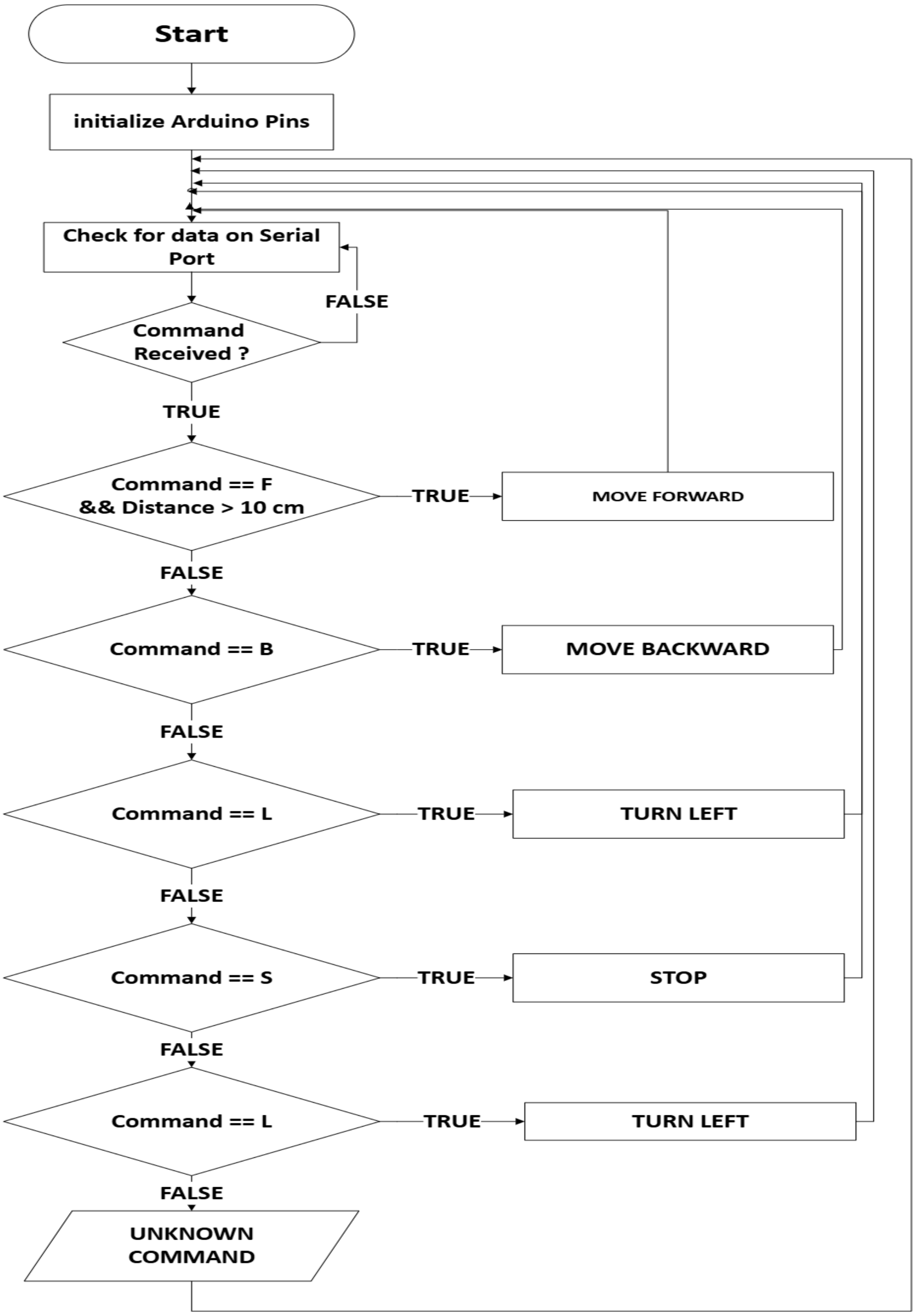

Figure 3.10 Flowchart representing the control logic of the wheelchair system.

3.2.2 Android Application Development and Debugging

Java is used as the programming language for creating an Android application with Android Studio. The app has voice command capabilities and directional control buttons. Designing the user interface, setting up Bluetooth, and verifying communication with the Arduino board are important development responsibilities. Android Studio's debugging tools are used to test data transmission and user interface responsiveness.

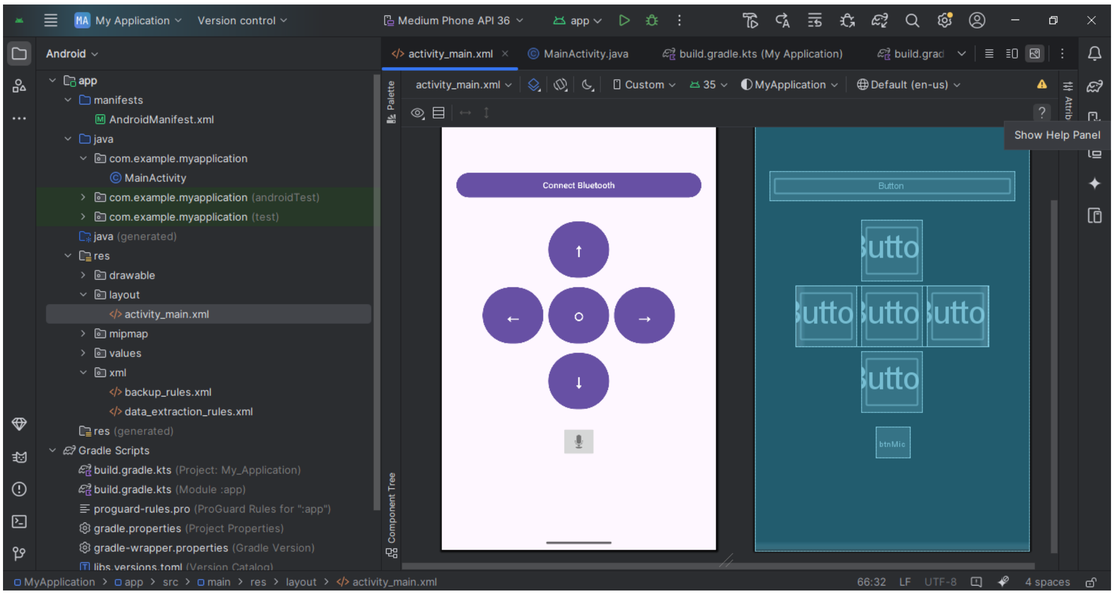

Figure 3.11 Android Studio

3.2.3 Integration and System level Testing

Once it has been confirmed that the Arduino and Android app operate separately, they are combined to form a whole system. To guarantee smooth hardware-software interaction, testing is done. This include verifying Bluetooth connectivity, evaluating the wheelchair's reactivity to voice and directional directions, and implementing any required modifications to enhance user experience and stability.

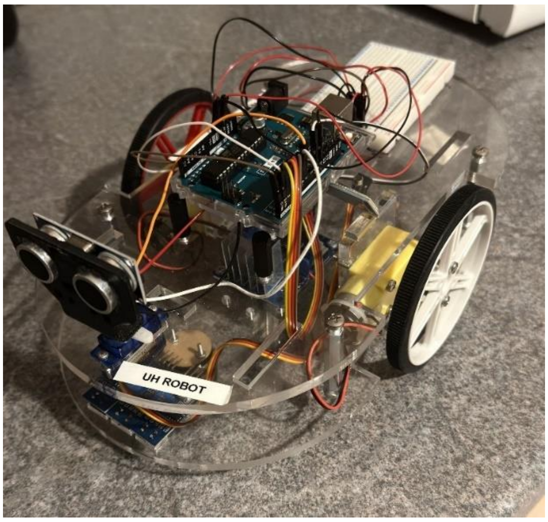

Figure 3.12 Complete System Setup

Chapter 4

RESULTS AND DISCUSSION

This chapter presents the results obtained after implementing the voice-controlled wheelchair system using a 2WD robot chassis, Arduino Uno, HC-05 Bluetooth module, ultrasonic sensor (HC-SR04), BO DC motors, and the custom-developed Android application. Each function of the system was tested to evaluate performance, responsiveness, and accuracy in various conditions.

4.1 Generation of PWM Signal on Arduino

Making a Pulse Width Modulation (PWM) signal on the Arduino was the first step. Proteus was used to mimic this signal in order to assess its usefulness. The oscilloscope in Proteus correctly displayed the created PWM signal, demonstrating that the Arduino could generate the required waveform for regulating the motor's speed.

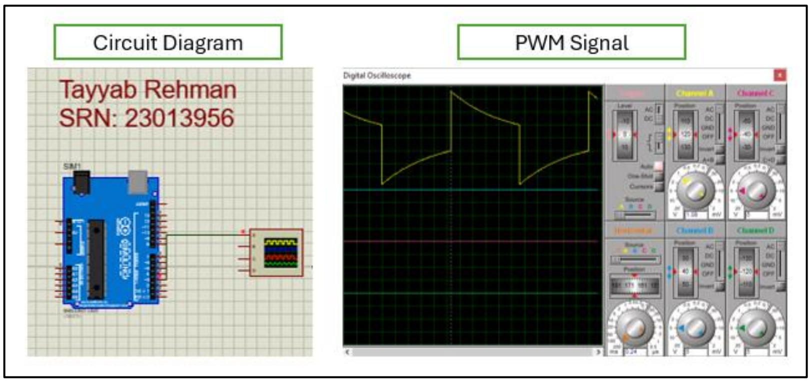

Figure 4.1 PWM Signal Generated by Arduino in Proteus

4.2 Integration and Testing of L298N Motor Driver

Following the Arduino's successful generation of the PWM signal, the L298N motor driver had to be integrated into the system. The L298N was selected because it can use the PWM signal from the Arduino to control the motor's speed and direction. To test the system, a DC motor was interfaced with the L298N and the motor driver was connected to the Arduino. After the integration was finished, the motor responded to the PWM signal by varying its speed in accordance with the PWM's duty cycle. Additionally, by altering the input signals to the L298N, the direction of the motor was successfully regulated. The system's expected performance validated that the motor control mechanism was operating as intended.

.

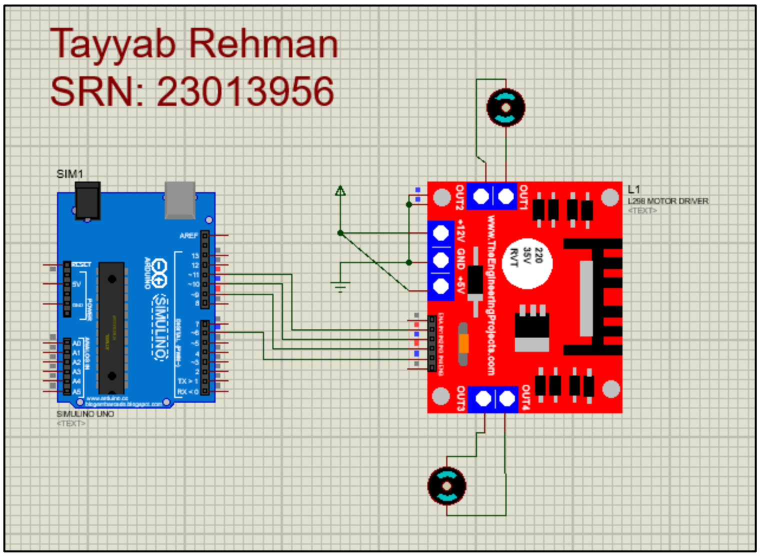

Figure 4.2 L298N Motor Driver Integration and Testing

4.3 Experimental Results

Once it has been confirmed that the Arduino and Android app operate separately, they are combined to form a whole system. To guarantee smooth hardware-software interaction, testing is done. This include verifying Bluetooth connectivity, evaluating the wheelchair's reactivity to voice and directional directions, and implementing any required modifications to enhance user experience and stability.

4.4.1 Voice Command and Joystick Control

The Android app correctly identified spoken orders like "forward," "left," "stop," and so on, and sent them via Bluetooth as single-character commands (like "F" and "L"). These commands were received and executed by the Arduino with little delay (~1-2 seconds). The on-screen joystick of the software allowed for seamless manual control. The Arduino demonstrated real-time response with precise directional changes by swiftly translating directional inputs into movement actions..

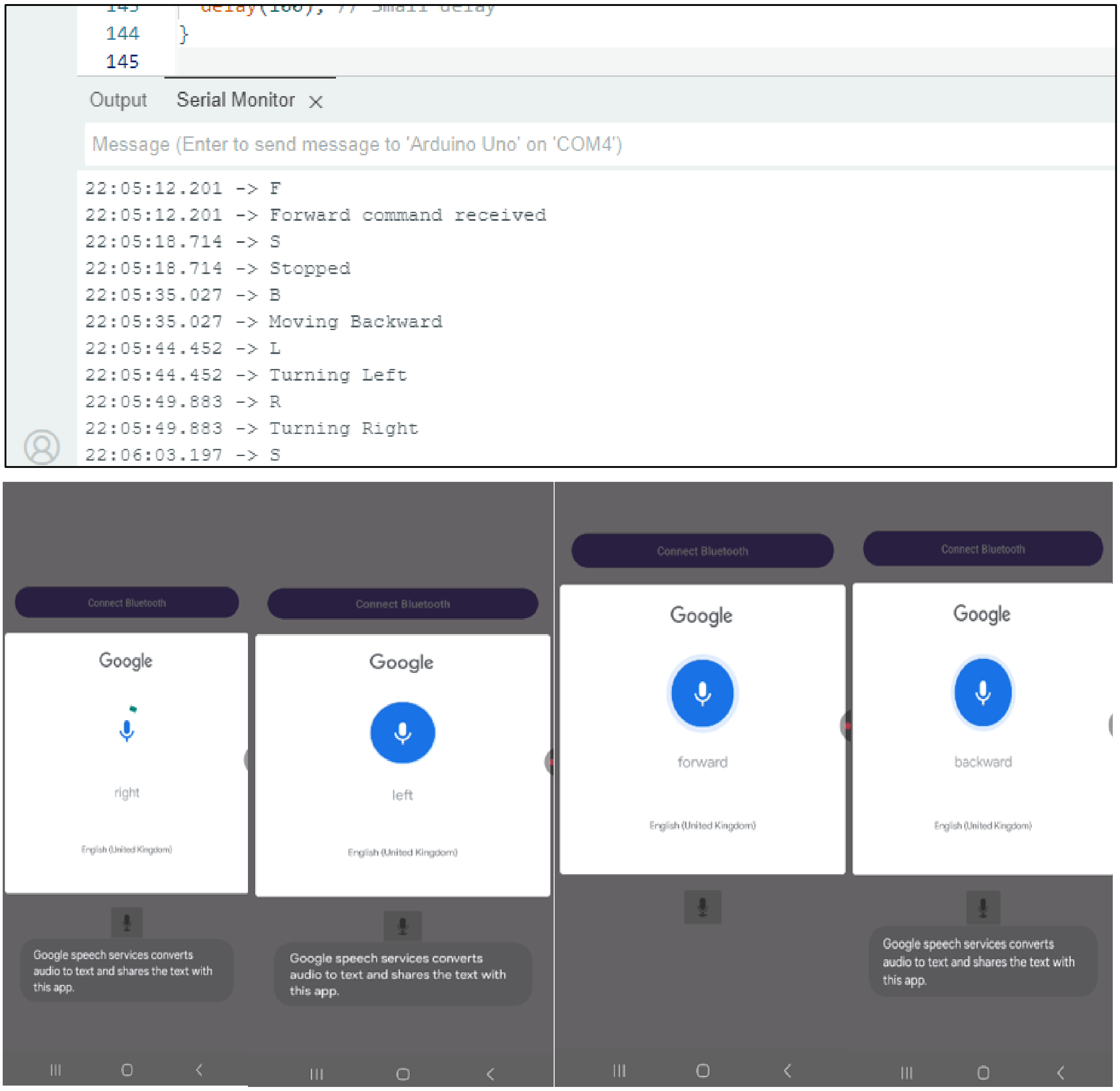

Figure 4.3 Voice Command Response via Bluetooth

4.4.2 Directional Movement of the Wheelchair

Basic directional movements, such as forward, backward, left, and right motions, were evaluated for the manual control system. Dedicated buttons controlled each direction, and the wheelchair's real-time reaction was guaranteed by Bluetooth communication.

The observed movements are depicted in the following diagrams:

The wheelchair travels straight ahead at a constant speed when the forward button is pressed.

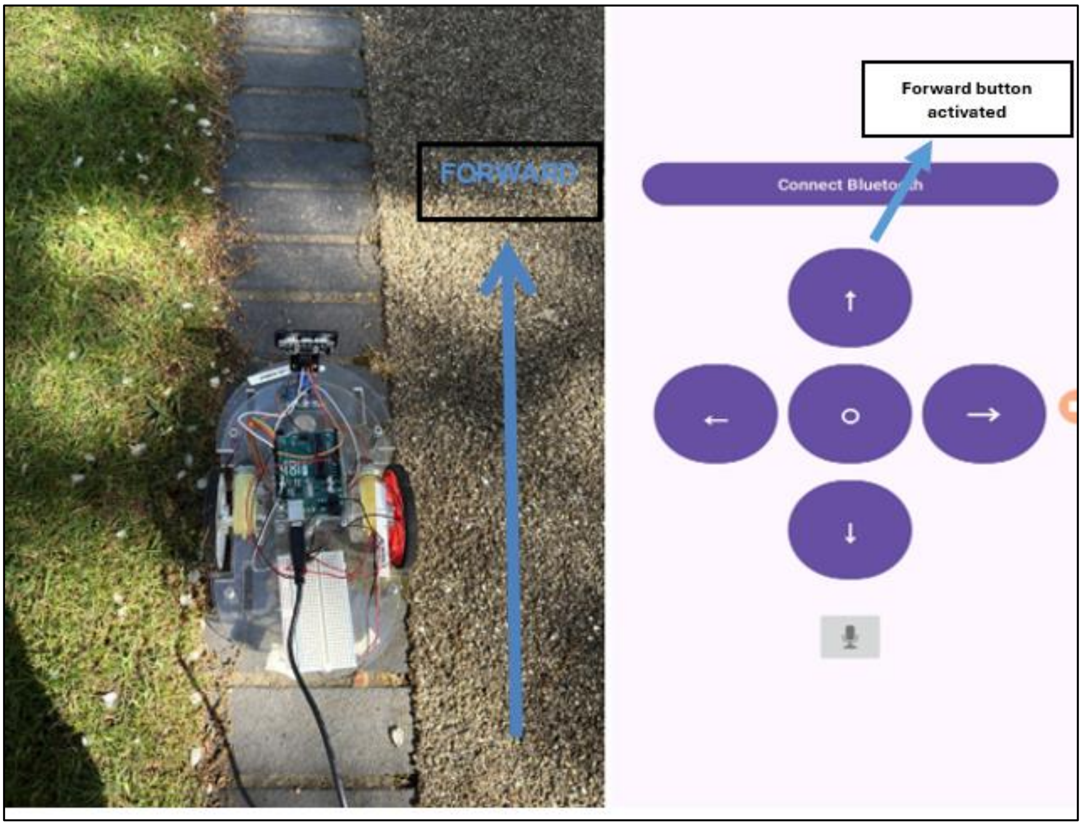

Figure 3.13 The wheelchair moves forward when the Forward button is pressed

Pressing the right button rotates the wheelchair to the right by slowing down the right wheel and speeding up the left wheel.

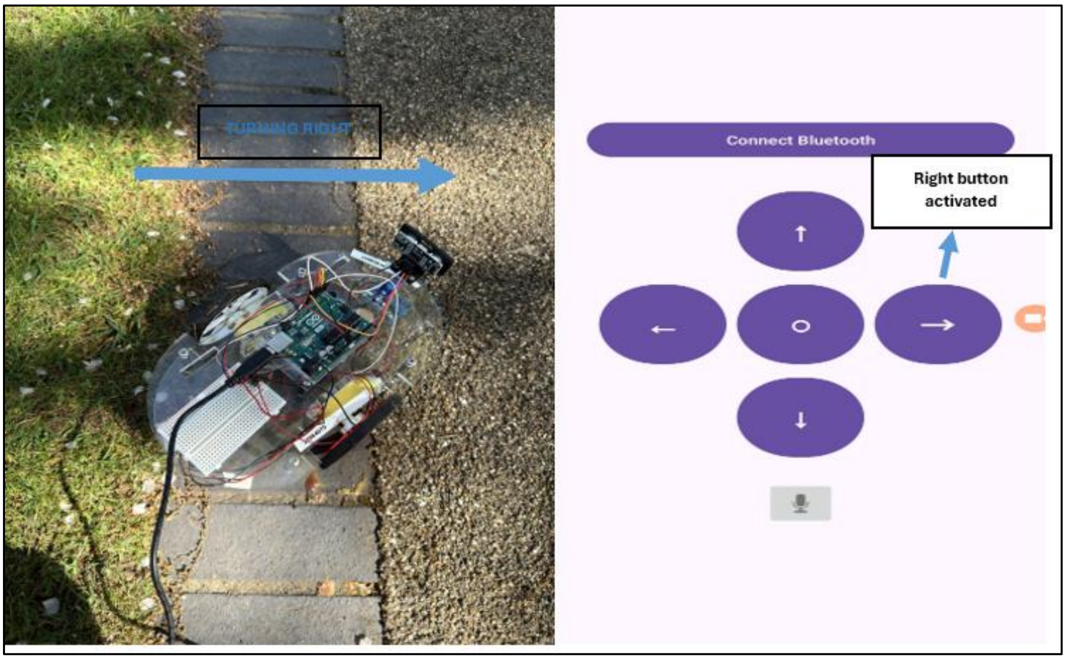

Figure 3.14 Wheelchair turning right on button press

Activating the left button rotates the wheelchair to the left by slowing down the left wheel and speeding up the right wheel.

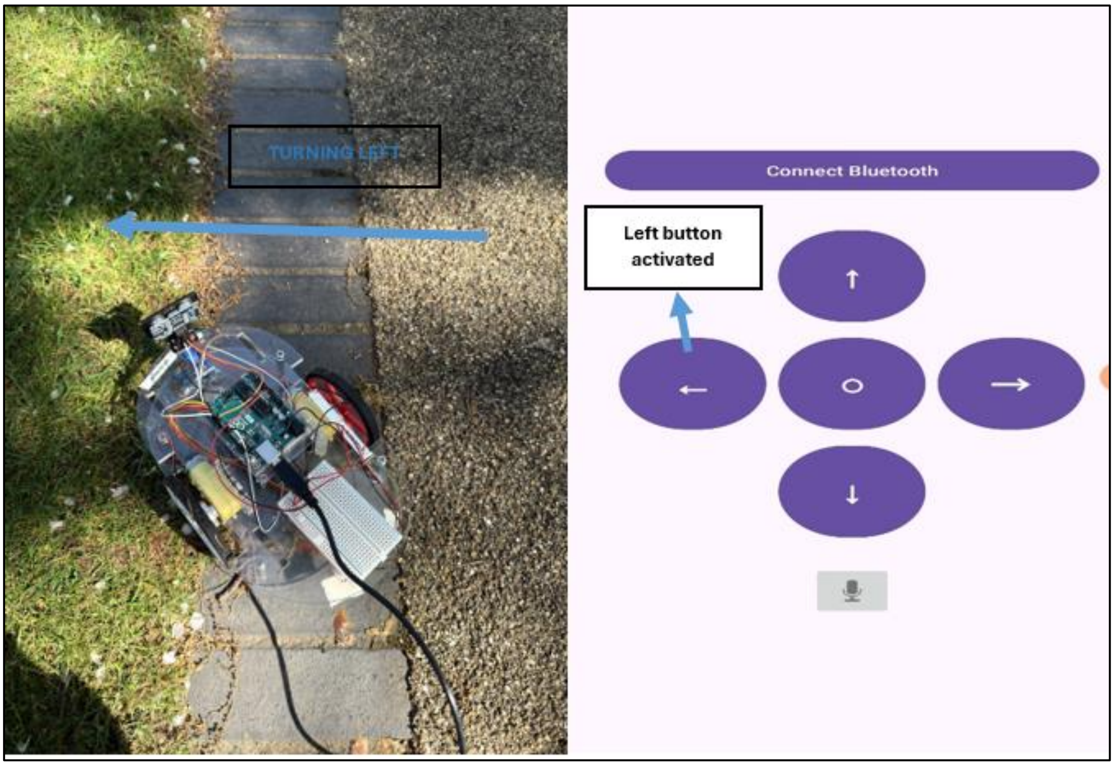

Figure 3.15 Wheelchair turning left on button press

4.4.3 Bluetooth Communication

Within a 10-meter range, the HC-05 Bluetooth module kept a steady connection. The connection status was accurately reflected by the app's status indicator. Throughout testing, there was no significant latency or data loss..

4.4.4 Obstacle Detection Using Ultrasonic Sensor

The ultrasonic sensor was configured to track the robot's distance from any obstructions in its route continually. During testing, the mobile application used Bluetooth to send a Forward command. The Serial Monitor output indicates that the Arduino successfully received this command.

The ultrasonic sensor picked up a nearby object as the robot advanced. The robot immediately halted to avoid a collision as soon as the measured distance fell below the 10-cm safety threshold. The following series of messages appeared on the Serial Monitor:

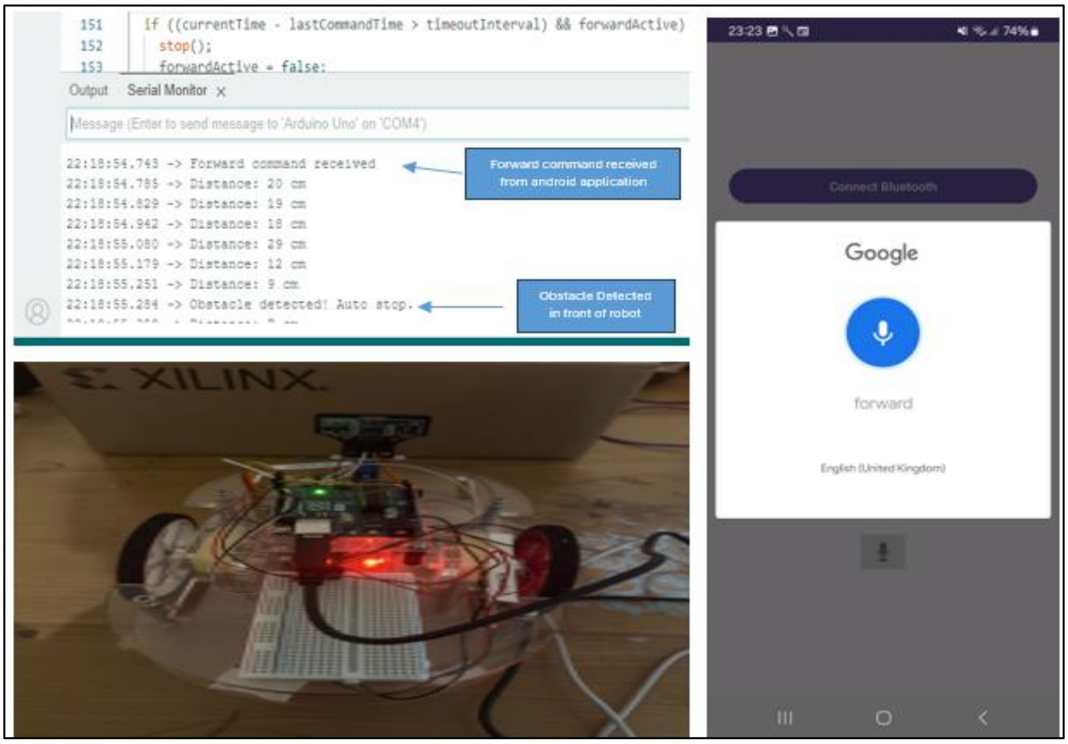

Figure 3.16 App-Controlled Robot Response

4.4.5 Limitations of the Project

While the project successfully demonstrated the concept of a voice-controlled robotic assistant integrated with a wheelchair, several limitations must be considered:

1. **Prototype Nature**: Comparing the developed system to a completely production-ready solution may reveal limitations in its functionality as it is a prototype. To guarantee robustness, dependability, and user safety, more engineering would be needed for the integration with an actual wheelchair.
2. **Voice Recognition Limitations**: This project's voice recognition system depends on Google's API, which might not be the best option for people with speech difficulties or in noisy settings. Due to the incomplete implementation of offline voice recognition methods, the system necessitates an active internet connection, which may be a drawback in some situations.
3. **Obstacle Detection and Navigation**: The precision and range of the ultrasonic sensor utilised for obstacle detection are limited, particularly in dynamic or complicated situations. For more accurate and dependable navigation, additional sensors like LIDAR or infrared are required, however this project did not incorporate them..
4. **Limited Sensor Integration**: Only simple ultrasonic sensors are used in the current system to identify obstacles. To travel more successfully in real-world settings, a true wheelchair system would need more sophisticated sensor systems (such as cameras and LIDAR), which was outside the purview of this experiment.
5. **Hardware Compatibility**: Compatibility issues may arise when the voice-activated technology is integrated with the motor and control system of an actual wheelchair. A

completely functional wheelchair system would require modifications to the current motor control circuits, power requirements, and safety measures.

6. **Testing Environment**: The controlled, indoor settings in which the testing was carried out may not accurately reflect the variety of settings in which a wheelchair might function in the actual world. To guarantee the system's resilience and dependability in practical situations, more testing in other contexts such as congested areas or the outdoors is required..
7. **User Customization**: Deep user customisation, including customised voice commands or modifications for certain disability, is not presently possible with the technology. These kinds of features would have to be included in a complete implementation in order to meet the demands of a wide range of users..

These restrictions draw attention to areas where the project might be improved and expanded upon in subsequent iterations. The key to moving from a working prototype to a fully functional system fit for practical application will be overcoming these obstacles..

Chapter 5

Conclusions, Recommendations and Project Management Review

5.1 Conclusions

With an emphasis on the value of assistive technology in helping people with mobility disabilities, this study presented the idea of a voice-activated wheelchair. In order to provide a practical and affordable solution for those with physical limitations, the goal was to build and prototype a wheelchair that uses speech recognition technology to control its movement. The literature review in Chapter 2 examined current research and technology related to robotic wheelchairs, voice-activated systems, and Arduino robots. This part established the current project's relevance and possible impact while highlighting the noteworthy developments in assistive technologies. Chapter 3's presentation of the system design listed the main project components. The Arduino Uno microcontroller, relay modules for motor movement control, DC motors for wheelchair operation, ultrasonic sensors for obstacle detection, and a voice recognition system driven by the Google API were among them. In order to ensure that the prototype could be used in a real-world situation without being overly complicated or expensive, the design also took into account pragmatic factors including cost-effectiveness and real-world usability. The procedure used to construct the voice-controlled wheelchair was explained in Chapter 4, covering every stage from wiring and hardware assembly to microcontroller programming and voice recognition system integration. Along with troubleshooting measures to address any difficulties that developed throughout the development process, it also detailed the testing procedures used to make sure the system worked as intended. The Arduino Uno microcontroller, relay modules for motor movement control, DC motors for wheelchair operation, ultrasonic sensors for obstacle detection, and a voice recognition system driven by the Google API were among them. In order to ensure that the prototype could be used in a real-world situation without being overly complicated or expensive, the design also took into account pragmatic factors including cost-effectiveness and real-world usability. The procedure used to construct the voice-controlled wheelchair was explained in Chapter 4, covering every stage from wiring and hardware assembly to microcontroller programming and voice recognition system integration. Along with troubleshooting measures to address any difficulties that developed throughout the development process, it also detailed the testing procedures used to make sure the system worked as intended. In order to improve the wheelchair's performance and dependability, this chapter also included suggestions for future developments, such as increasing the precision of voice command recognition, adding more sensors for improved navigation, and investigating more sophisticated motor control systems. The project's findings demonstrate its promise as a viable assistive mobility solution, and more improvements could make it even more useful and efficient for people with mobility issues.

5.2 Recommendations

Based on the successful implementation and testing of the voice-controlled robotic assistant, several recommendations are proposed for future development:

- **Enhanced Voice Recognition**

While the system currently relies on Google's voice recognition API, integrating an offline voice recognition solution could improve system reliability in environments with limited internet access. Exploring local voice processing options would reduce dependence on internet connectivity.

- **Improved Obstacle Detection**

Although the ultrasonic sensor works well for detecting nearby objects, adding additional sensors, such as infrared or LIDAR, could enhance the robot's ability to navigate more complex environments with greater accuracy.

- **Navigation System**

Integrating a more advanced navigation system, such as a path-planning algorithm, could enable the robot to autonomously navigate a room or environment. This would involve adding sensors to track its position and avoid obstacles while following a predefined path, improving its autonomy and usability in different settings.

- **Wheelchair Integration**:

It is recommended that the project be moved from the prototype phase to real-world testing. This should begin with the integration of the voice-controlled robotic assistant into a real wheelchair, incorporating safety features and additional sensor technologies for navigation and obstacle detection.

References

[1] ChatGPT, "AI-driven Voice-Controlled Wheelchair Illustration," OpenAI, 27 February 2025. [Online]. Available: OpenAI ChatGPT platform. [Accessed 27 February 2025].

[2] J. Leaman, "A Comprehensive Review of Smart Wheelchairs: Past, Present, and Future," *IEEE Transactions on Human-Machine Systems,* vol. 47, no. 4, pp. 486-499, 2017.

[3] M. A. S. Ahmed, "Design, Modeling, And Implementation of PIC Based Electrical Wheelchair," *Diyala Journal of Engineering Sciences,* vol. 7, no. 4, pp. 62-81, 2024.

[4] M. A. Hossain, "IoT Based Smart Wheelchair for Disabled People," *Proceedings of the 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI),* pp. 1-5, 2021.

[5] M. Price, "Rehabmart," Rehabmart, [Online]. Available: https://www.rehabmart.com/post/7-tips-to-help-you-choose-the-best-wheelchair. [Accessed 25 02 2025].

[6] "Clinical resources," 04 05 2017. [Online]. Available: https://aci.health.nsw.gov.au/networks/spinal-cord-injury/spinal-seating/module-10/keep-the-big-picture-in-mind. [Accessed 25 02 2025].

[7] Q. S. C. I. Service, "Queensland Spinal Cord Injury Service," October 2015. [Online]. Available: https://www.health.qld.gov.au/qscis/documents/pdwc-features.pdf. [Accessed 25 02 2025].

[8] H. Ismail, R. Roy and L.-J. Sheu, "Exploration-Based SLAM (e-SLAM) for the Indoor Mobile Robot Using Lidar," *Sensors,* vol. 22, no. 4, p. 1689, 2022.

[9] G. G. Goyer and R. Watson, "The Laser and its Application to Meteorology," *Bulletin of the American Meteorological Society,* vol. 44, no. 9, pp. 564-570, 1963.

[10] M. Nishimori, S. Takeshi and R. Konishi, "Voice Controlled Intelligent Wheelchair," in *SICE Annual Conference 2007*, 2007.

[11] H. Kian , Yagasena and Chelladurai, "Arduino based voice controlled wheelchair," *Journal of Physics: Conference Series,* vol. 1432, no. 1, 2020.

[12] "Arduino UNO - JavaTpoint - Tpoint Tech," [Online]. Available: https://www.tpointtech.com/arduino-uno. [Accessed 25 02 2025].

[13] "Components101," [Online]. Available: https://components101.com/modules/l293n-motor-driver-module. [Accessed 25 02 2025].

[14] "Electro Boat," [Online]. Available: https://electroboat.in/products/details/bo-motor-60-rpm-single-shaft. [Accessed 25 02 2025].

[15] "Kitronik Ltd," [Online]. Available: https://kitronik.co.uk/products/46130-ultrasonic-distance-sensor-hc-sr04-5v-version. [Accessed 22 4 2025].

[16] “ROBO CY,” [Online]. Available: https://robo.com.cy/products/bluetooth-hc-05-ble-module. [Accessed 22 4 2025].

[17] “Amazon,” [Online]. Available: https://www.amazon.co.uk/DollaTek-Battery-Chassis-Encoder-Arduino. [Accessed 28 04 2025].

[18] DroneBot Workshop, [Online]. Available: https://dronebotworkshop.com/arduino-microcontroller/. [Accessed 25 02 2025].

[19] A. Aqeel, “The Engineering Projects,” 19 04 2018. [Online]. Available: https://www.theengineeringprojects.com/2018/04/introduction-to-raspberry-pi-3.html. [Accessed 25 02 2025].

[20] “The Pi Hut,” [Online]. Available: https://thepihut.com/products/tfmini-s-lidar-ranging-sensor. [Accessed 25 02 2025].

[21] “BOTLAND,” [Online]. Available: https://botland.store/gravity-sound-modules/23332-gravity-offline-language-learning-voice-recognition-sensor-i2c-uart-dfrobot-sen0539-en-6959420923342.html. [Accessed 25 02 2025].

[22] J. Marquis, “Mobility Plus Colorado,” 21 4 2022. [Online]. Available: https://www.mobilitypluscolorado.com/blog/how-does-electric-wheelchair-work. [Accessed 25 2 2025].

[23] ChatGPT, “Voice Recognition Methods,” OpenAI, [Online]. Available: https://chat.openai.com. [Accessed 3 March 2025].

APPENDICES

Appendix A: Arduino Code for Wheelchair Control

```
#include <SoftwareSerial.h>

#define echoPin 6
#define trigPin 7
long duration;
int distance;
bool forwardActive = false;  // Track if moving forward

// Motor control functions
void movefwd() {
  analogWrite(10, 200);
  analogWrite(11, 200);
  digitalWrite(2, 0);
  digitalWrite(3, 1);
  digitalWrite(5, 0);
  digitalWrite(4, 1);
}

void moveback() {
  analogWrite(10, 200);
  analogWrite(11, 200);
  digitalWrite(2, 1);
  digitalWrite(3, 0);
  digitalWrite(5, 1);
  digitalWrite(4, 0);
}

void stop() {
  analogWrite(10, 0);
  analogWrite(11, 0);
  digitalWrite(5, 0);
  digitalWrite(4, 0);
  digitalWrite(2, 0);
  digitalWrite(3, 0);
}

void left_turn() {
  analogWrite(10, 200);
  analogWrite(11, 0);
  digitalWrite(2, 0);
  digitalWrite(3, 1);
  digitalWrite(5, 0);
  digitalWrite(4, 1);
}
```

```
void right_turn() {
  analogWrite(10, 0);
  analogWrite(11, 200);
  digitalWrite(2, 0);
  digitalWrite(3, 1);
  digitalWrite(5, 0);
  digitalWrite(4, 1);
}

// Get distance from ultrasonic sensor
int getDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  return distance;
}

// Setup function
void setup() {
  pinMode(11, OUTPUT);  // Left motor enable PWM
  pinMode(10, OUTPUT);  // Right motor enable PWM
  pinMode(3, OUTPUT);   // INA1
  pinMode(4, OUTPUT);   // INA2
  pinMode(5, OUTPUT);   // INA3
  pinMode(2, OUTPUT);   // INA4
  pinMode(trigPin, OUTPUT);  // Trig pin
  pinMode(echoPin, INPUT);   // Echo pin

  Serial.begin(9600);
  stop(); // Stop at start
}

// Main loop
void loop() {
  // Check serial input
  if (Serial.available() > 0) {
    char command = Serial.read();
    Serial.println(command);

    switch (command) {
      case 'F':
        forwardActive = true;
        Serial.println("Forward command received");
```

```
        break;

      case 'B':
        moveback();
        forwardActive = false;
        Serial.println("Moving Backward");
        break;

      case 'L':
        left_turn();
        forwardActive = false;
        Serial.println("Turning Left");
        break;

      case 'R':
        right_turn();
        forwardActive = false;
        Serial.println("Turning Right");
        break;

      case 'S':
        stop();
        forwardActive = false;
        Serial.println("Stopped");
        break;

      default:
        stop();
        forwardActive = false;
        Serial.println("Unknown command. Motors stopped.");
        break;
    }
  }

  // Always monitor distance
  int dist = getDistance();

  // Handle forward mode with distance safety
  if (forwardActive) {
    if (dist > 10) {
      movefwd();
    } else {
      stop();
      forwardActive = false;
      Serial.println("Obstacle detected! Auto stop.");
    }
  }
```

```
  delay(100); // Small delay
}
```